



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Přeprogramování UI aplikace Kytarový zpěvník s ohledem na UX
<b>Student:</b>	Nikolai Baranov
<b>Vedoucí:</b>	Ing. Zdeněk Rybala
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2018/19

### Pokyny pro vypracování

Cílem bakalářské práce je přeprogramování uživatelského rozhraní aplikace Kytarový zpěvník v rámci změny architektury na MVP a s ohledem na UX (User eXperience). Zároveň bude UI doplněno o nové prvky a funkce dle požadavků vedoucího práce.

Souběžně s touto prací bude probíhat v rámci bakalářské práce dalšího studenta návrh a realizace změny architektury aplikace na MVP.

Pokyny k vypracování:

- popište stávající řešení UI aplikace Kytarový zpěvník
- popište principy UX týkající se funkcí aplikace Kytarový zpěvník
- proveďte analýzu požadavků na ovládání aplikace, včetně nově stanovených funkčních požadavků a v návaznosti na změnu architektury aplikace na MVP
- proveďte návrh řešení potřebných změn a návrh patřičných UI prvků a obrazovek
- realizujte navržené změny, vhodným způsobem otestujte a zdokumentujte
- vytvořte uživatelskou příručku a ve spolupráci s paralelní prací vytvořte programátorskou příručku

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 28. ledna 2018





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Přepřacování UI aplikace Kytarový zpěvník s ohledem na UX**

*Nikolai Baranov*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Zdeněk Rybola, Ph.D

26. června 2018



---

## Poděkování

V první řadě bych chtěl poděkovat Ing. Zdeňku Rybolovi, Ph.D, vedoucímu mé bakalářské práce, za odborné vedení a konzultace, trpělivost a čas, který mi v průběhu zpracování bakalářské práce věnoval. Velké poděkování náleží rovněž mému kamarádovi a kolegovi Janu Holanovi.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 26. června 2018

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2018 Nikolai Baranov. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Baranov, Nikolai. *Přepřpracování UI aplikace Kytarový zpěvník s ohledem na UX*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.



---

# Abstrakt

Cílem této práce je zanalyzovat stávající řešení uživatelského rozhraní webové aplikace Kytarový zpěvník. Na základě výsledků analýzy navrhnout možnosti vylepšení. Hlavní důraz při návrhu bude kladen na použití technik User Experience, které musí usnadnit hraní na kytaru přes tuto aplikaci. V souladu s návrhem vytvořit a implementovat nové uživatelské rozhraní. Dále je součástí práce analýza, návrh a implementace nových funkčních požadavků, které budou realizovány ve spolupráci s kolegou, který v rámci své závěrečné práce přepracuje tuto aplikaci do MVP architektury.

**Klíčová slova** kytarový zpěvník, uživatelské rozhraní, uživatelská přívětivost, UI, UX, design, JS, CSS



---

# Abstract

The aim of this work is to analyze the existing user interface of „Guitar Songbook“ web application. Based on the results of the analysis, suggest improvements. The main emphasis will be put on using the User Experience technologies, which must make easier to play the guitar through this application. In accordance with the proposal, create and implement a new user interface. Also a part of this work is analysis, design and implementation of new functional requirements that will be implemented in cooperation with a colleague who will rework this application into the MVP architecture.

**Keywords** guitar songbook, user interface, user experience, UI, UX, design, JS, CSS



---

# Obsah

<b>Úvod</b>	<b>1</b>
Cíl práce . . . . .	2
<b>1 Analýza</b>	<b>3</b>
1.1 Úvod do problematiky User Experience . . . . .	3
1.2 Analýza struktury a navigace webu . . . . .	8
1.3 Analýza procesů v aplikaci . . . . .	8
1.4 Funkční požadavky . . . . .	12
1.5 Klíčové aspekty při navrhování UI z hlediska UX . . . . .	13
1.6 Architektura aplikace a použité technologie . . . . .	15
<b>2 Návrh</b>	<b>19</b>
2.1 Architektura . . . . .	19
2.2 Vytvoření prototypu a návrh designu . . . . .	19
2.3 Návrh řešení požadavků, které vznikly při analýze UI z pohledu UX . . . . .	20
2.4 Adresářová struktura projektu . . . . .	23
<b>3 Realizace</b>	<b>25</b>
3.1 Použité technologie . . . . .	25
3.2 Implementace . . . . .	28
3.3 Testování . . . . .	30
3.4 Nasazení . . . . .	31
<b>Závěr</b>	<b>33</b>
<b>Literatura</b>	<b>35</b>
<b>A Seznam použitých zkratk</b>	<b>39</b>



---

## Seznam obrázků

1.1	Model „včelího plástu“ Petera Morvilla . . . . .	5
1.2	Model na základě spektra . . . . .	6
1.3	Pětiúrovňový model Jesse Garretta . . . . .	7
1.4	Informační architekturu aplikace . . . . .	9
1.5	Obchodní proces hraní písničky . . . . .	11
1.6	Obchodní proces importovat . . . . .	12
1.7	Model nasazení . . . . .	17
2.1	Prototyp hlavní stránky webu . . . . .	20
2.2	F-pattern šablona . . . . .	21
2.3	Nový logotyp aplikace . . . . .	22
2.4	Struktura projektu . . . . .	23
3.1	Dialogové okno pro export zpěvníku . . . . .	29
3.2	Princip činnosti Selenium WebDriver . . . . .	32





---

# Úvod

Internet je v současnosti jedním z neaktivněji se rozvíjejících zdrojů informací. V éře high-tech technologií World Wide Web (WWW) představuje intelektuální vymoženost lidstva. Dnešní WWW pozůstává z milionů webů, na nichž jsou zpřístupněny nejrůznější informace.

Webové stránky podporující technologii multimédií zahrnují v sobě různé typy poskytování informací - prostřednictvím textu, grafiky, zvuku, animace či videa. Od toho, jak kvalitně a atraktivně je webová stránka vytvořena, ve velké míře závisí i její úspěch v síti.

Uživatel rád navštíví webové stránky vyznačující se stylovým designem, které nejsou zatíženy nadměrnou grafikou a animací, které se rychle spouštějí a v okně webového prohlížeče se správně zobrazují. Nicméně, uživatel je mnohem náročnější. Důležitou součástí úspěšnosti webové stránky představuje její celkový uživatelský prožitek. K vyřešení této záležitosti (co se týká tvorby webových stránek) byla vytvořena celá řada technik soustředěných v User Experience.

Profesionálně vytvořená webová stránka sama podporuje její vyhledávání uskutečňující se na základě zadání požadavku do vyhledávačů; cílové publikum totiž je tvořeno uživateli, kteří hledají na internetu konkrétní informace.

Internet umožňuje nejen informace získávat, ale taktéž sdílet je. Základní principy aplikace Kytarový zpěvník vycházejí z této myšlenky.

Webová aplikace Kytarový Zpěvník je databáze písní s akordy na kytaru. Hlavním cílem aplikace je usnadnění správy písniček pro kytaristy.

Moje seznámení s aplikací se údalo již před rokem, kdy jsme si já, kolega Jan Holan a ještě několik dalších studentů danou problematiku vybrali jako téma v rámci projektu vycházejícího z předmětů Softwarový týmový projekt 1 a Softwarový týmový projekt 2 bakalářského programu Fakulty informačních technologií ČVUT.

Pod vedením pana Ing. Zdeňka Ryboly, Ph.D se nám podařilo vytvořit tým, jehož každý účastník měl zřetelně stanovenou odpovědnost za určitou

část projektu. Díky tomuto jsem měl štěstí naplno se ponořit do vývoje frontend části aplikace. Na konci své účasti na projektu jsem jasně pochopil všechny jeho výhody a nevýhody. V průběhu diskuzí o dalším vývoji projektu jsme dospěli k poznání, že na základě projektu, jehož vývoj započal v roce 2014, se z malé webové stránky vyvinul multifunkční portál.

F. Brooks ve své knize *The Mythical Man-Month* psal: „In most projects, the first system built is barely usable....Hence plan to throw one away; you will, anyhow.“ [1]

Tato slova dokonalým způsobem charakterizují situaci, ve které jsme se ocitli. Rozvoj projektu vyžadoval promyšlenější architekturu, uplatnění více nových technologií a vytvoření podmínek, které by opětovný vznik této situace zamezovaly. Proto bylo rozhodnuto, že spolu s Janem Holanem přepracujeme aplikaci v rámci svých bakalářských prací. Činnost jsme si rozdělili podle rolí, které jsme měli předtím. Já se budu věnovat implementaci frontendové části a kolega Jan backendové části.

## Cíl práce

Hlavní účel rešeršní části je obeznámení čitatele se základními termíny a metodami, které se při vývoji současných webových stránek používají. Na základě prezentovaných technik provést analýzu aktuálního uživatelského rozhraní a následně vytvořit prototypy nových webových stránek. Kromě toho, za účelem vytvoření celostního obrazu provést analýzu aplikace jako celku, a to včetně nových požadavků, které budou v rámci této bakalářské práce realizovány. Další etapu následující po analýze představuje návrh a výběr potřebných technologií, které jsou pro rozvoj nevyhnutelné. A konečně, na základě výsledků získaných z provedených studií uskutečnit závěrečnou část – realizaci a testování.

---

# Analýza

V této části je popsána analýza webové aplikace Kytarový zpěvník. Nejdříve, pro správné pochopení obsahu této kapitoly jsou definovány metody, kterými bude aplikace analyzována. Poté je v dalších oddílech analyzován stav, v jakém se aplikace Kytarový zpěvník v době analýzy nachází a jaké nové funkčnosti budou v rámci této bakalářské práce realizovány.

## 1.1 Úvod do problematiky User Experience

### 1.1.1 Definice

Termín User Experience (UX) vytvořil počátkem devadesátých let 20. století kognitivní vědec Don Norman. Za účelem objasnění původu termínu Norman psal:

„Vytvořil jsem termín, protože vymezení jako rozhraní člověk-počítač (Human Computer Interaction, HCI) anebo použitelnost (usability) se mi zdálo být příliš plytké. Já jsem chtěl uchopit všechny aspekty uživatelských prožitků, které vyplývají ze vzájemné interakce mezi uživatelem a systémem, včetně průmyslového designu, grafiky, uživatelského rozhraní, fyzické interakce a vedení k činnosti.“[2]

V roce 2010 Mezinárodní organizace pro standardizaci (International Organization for Standardization, ISO) pojem User Experience oficiálně standardizovala. Podle ISO 9241-210[3] UX označuje „uživatelský prožitek“ (zaměření na potřeby spotřebitele) a reakci jednotlivce na základě použití nebo předpokládaného použití výrobku, systému nebo služby.

Termín UX je téměř vždy doprovázen termínem „design“. Je důležité upřesnit, že termín design v tomto kontextu znamená navrhování, projektování či projekci a nemá nic společného s grafickým designem.

User Experience Design (UxD) označuje komplexní přístup, co se týká vzájemné interakce mezi uživatelem a rozhraním webu, mobilní aplikace či

programu. Nabádá k vývoji co nejvhodnějšího a nejvíce uchopitelného produktu, jenž je určen pro určitou cílovou skupinu.

UX design je zodpovědný za to, jak produkt funguje a jaké prožitky v uživateli vyvolává. Zahrnuje v sobě tradiční navrhování interakcí HCI a všechny aspekty produktu nebo služby, které jsou vnímány uživatelem.

Termín UX design se v praxi mylně zaměňuje s termínem použitelnost (usability). Tento omyl je pochopitelný – použitelnost taktéž vyžaduje, aby produkt byl užitečný a snadno použitelný.

Vyskytuje se taktéž názor, že UX a uživatelské rozhraní (User Interface, UI) označují jedno a to samé; anebo tvrzení, že UX představuje čistě navrhování uživatelského rozhraní. To je taktéž chybné tvrzení.

User Interface (UI) doslovně znamená uživatelské rozhraní. UI se týká akumulace přístupů a prvků, jež uživateli umožňují interakci se systémem. Netýká se takových podrobností, jako je například reakce uživatele na systém.

### 1.1.2 Přehled existujících UX technik

Při navrhování UX lze využít bezpočet metodik a přístupů, z nichž každý má svoje výhody a nevýhody. Na základě studia různých přístupů k vývoji UX byly následující modely označeny za nejužitečnější:

1. Model „včelího plástu“ Petera Morvilla[4]
2. Model na základě spektra[5]
3. Pětiúrovňový model Jesse Garretta[6]

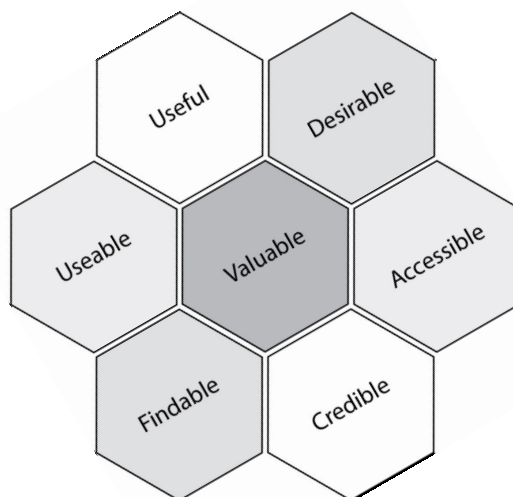
#### 1.1.2.1 Model „včelího plástu“ Petera Morvilla

Tento model představuje 7 interakčních oblastí (viz. obrázek 1.1), které kvalitativním způsobem určují UX.

V centru pozornosti vývoje produktu podle tohoto modelu stojí hlavní kritérium, a tím je hodnota produktu. Zákazník vyžaduje záruku toho, aby jeho produkt byl pro potenciálního uživatele cenný a oplýval taktéž dostatečnou konkurenční schopností ve srovnání s řadou podobných aplikací. Aby byl dosažen příznivý konečný výsledek, je nezbytné zajistit, aby v průběhu celého procesu vývoje byla věnována zvýšená pozornost všem kritériím souvisejícím s hlavním kritériem, a aby byla pohotově prověřována jejich aktuálnost.

Kromě souvztažnosti mezi všemi sousedícími kritérii a hlavním kritériem je důležité nepřestat věnovat pozornost tomu, jak se mezi sebou jednotlivá kritéria nacházející se ve vnějším okruhu modelu ovlivňují. Jsou uspořádány tak, že na každý atribut působí navzájem sousedící kritéria, z nichž každé je nezbytné, ale není postačující. Proto by měl být tento model používán pouze v jeho celistvosti, to znamená, že nelze opomenout ani jednu z jeho buněk.

Za výhodu může být považována skutečnost, že vyvíjený uživatelský prožitek prostřednictvím jednotlivých buněk (jedna po druhé), je snadné analyzovat,



Obrázek 1.1: Model „včelího plástu“ Petera Morvilla[7]

zda je vše provedeno tak, aby bylo jisté, že konečný výsledek produktu bude představovat hodnotu požadovanou zákazníkem.

### 1.1.2.2 Model na základě spektra

Ústřední prvek tohoto modelu představují jednotlivci (nebo celé týmy jednotlivců), bez nichž není možné uživatelský prožitek navrhnout – vývojáři softwaru, návrháři a zákazníci. Na obrázku 1.2 lze zřetelně vyzorovat souvislost mezi jednotlivými týmy v každé vývojové fázi. Například by měli spolupracovat inženýři a návrháři z toho důvodu, aby rozhraní produktu bylo co nejjednodušší a uživatelsky co nejkomfortnější, nebo všechny týmy by měly přispět k tomu, aby celý produkt byl konzistentní. Největší výhodou tohoto modelu představuje skutečnost, že lze velmi snadno provést kontrolu. Každá osoba podílející se na vývoji si uvědomuje svou vlastní odpovědnost a odpovědnost těch, s nimiž se musí, za účelem provedení práce, nezbytně kontaktovat. To zároveň představuje největší nevýhodu. Někteří jednotlivci nevyhledávají kontakt s ostatními, a nejvíc ze všeho nemají rádi vzájemný kontakt na základě předložených instrukcí. Vzhledem k tomu, že zákazníci vždycky nemohou jasně a přesně definovat své požadavky, anebo nemají dostatečné povědomí o tom, co můžou požadovat, a tak technici jejich požadavky nemohou splnit. V důsledku toho trpí celý projekt.



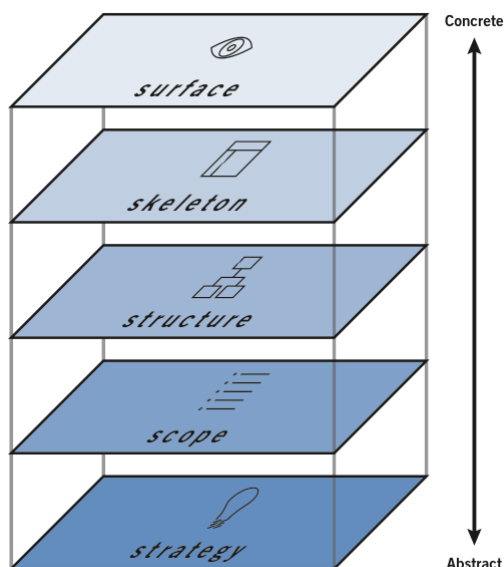
Obrázek 1.2: Model na základě spektra[8]

### 1.1.2.3 Pětiúrovňový model Jesse Garretta

Model uvedený na obrázku 1.3 představuje integrovaný přístup k navrhování uživatelského prožitku.

Model Jesse Garretta pozůstává z pěti úrovní, které umožňují snadno rozvíjet UX od abstraktní myšlenky, až po konkrétní produkt.

- **Úroveň strategie.** Na této úrovni jsou řešeny cíle aplikace a to, co by při práci s ní měli uživatelé obdržet. Důležitou součástí této úrovně je shromažďování informací, to znamená zhodnocení všech potřeb a zdrojů informací, a to spolu se zákazníkem.
- **Úroveň souboru možností.** Na této úrovni je nevyhnutelné shrnout seznam všech funkcí projektu a požadavků na obsah, a shrnout specifčnosti funkčnosti a požadavky, které jsou potřebné k dosažení strategických cílů
- **Úroveň struktury.** Na základě údajů získaných na předchozích úrovních se vytváří celkový obraz vývoje. Důležitou etapu této úrovně představuje výběr informační architektury (Information Architecture, IA).
- **Úroveň kostry.** Na této úrovni se věnuje pozornost třem typům navrhování: navrhování rozhraní, navrhování navigace a navrhování shromažďování informací.



Obrázek 1.3: Pětiúrovňový model Jesse Garretta[9]

- **Úroveň vzhledu.** Tato úroveň odráží styl vizuální komunikace firemní značky a přitahuje se pozornost k důležitým prvkům na základě jejich zjevného vyčlenění ve srovnání s ostatními informacemi.

Ve srovnání s ostatními prezentovanými modely je pro výše zmíněný model typický jasný a zřetelně strukturovaný přístup k rozvoji informačního prostoru a softwarového prostředí jako celku. Taktéž za výhodu tohoto modelu se považuje možnost úplného a detailního rozvoje, co se uživatelského prožitku týká, od strategie až po vizuální design. Za nedostatek se považuje skutečnost, že pro vývoj uživatelského prožitku podle tohoto modelu je potřeba mnoho času.

### 1.1.3 Strategie při vývoji aplikace Kytarový Zpěvník

Při analýze i projektování aplikace Kytarový Zpěvník byl použit Garrettův model. Jak již bylo zmíněno v předchozím oddílu 1.1.2.3, proces navrhování v tomto případě je rozdělen do následujících úrovní:

- **Úroveň strategie.** Tato úroveň je nejvíce abstraktní a nejvíce koncepční. Hlavním cílem, na této úrovni, bylo zjistit, jaké úkoly mohou uživatelé prostřednictvím aplikace řešit. Za tímto účelem bylo nezbytné provést analýzu jak základních existujících, tak i nových obchodních procesů. Této analýze je věnován oddíl 1.3

- **Úroveň souboru možností.** Aby tato úroveň byla uchopena v celé své plnosti, byly zavedeny nové funkční požadavky, které jsou popsány v oddílu 1.4.
- **Úroveň struktury.** Hlavním úkolem, který byl v této fázi stanoven, je popsat současnou informační architekturu a posoudit potřebu přechodu na novou. Prozkoumání IA se nachází v sekci 1.3.2
- **Úroveň kostry.** Tato úroveň představuje přechod od abstraktního pohledu na produkt k jeho konkrétnímu provedení. Na této úrovni byla provedena analýza uživatelského prožitku s větším důrazem na zaměření se na obecný dojem produktu. Analýze uživatelského prožitku aplikace Kytarový zpěvník je věnována kapitola 1.5. V této fázi projektu byly také vytvořeny prototypy. Vytvoření prototypů je zaznamenáno v oddílu 2.2.
- **Úroveň vzhledu.** Tato úroveň je konečná. Hlavním úkolem bylo navrhnout vizuální úpravu a přejít od prototypu k vytvoření konečného řešení. Tomu je věnována kapitola 3

### 1.2 Analýza struktury a navigace webu

Termín informační architektura označuje strukturu všech informací umístěných na webu – uspořádání stránek podle různých souvislostí do jednotlivých kategorií. Informační architektura je neviditelná. Pro zjištění správnosti IA je potřeba nakreslit mapu stránek. Obrázek 1.4 udává informační architekturu aplikace. Současný stav struktury odpovídá nárokům UX a projevuje se logickým způsobem:

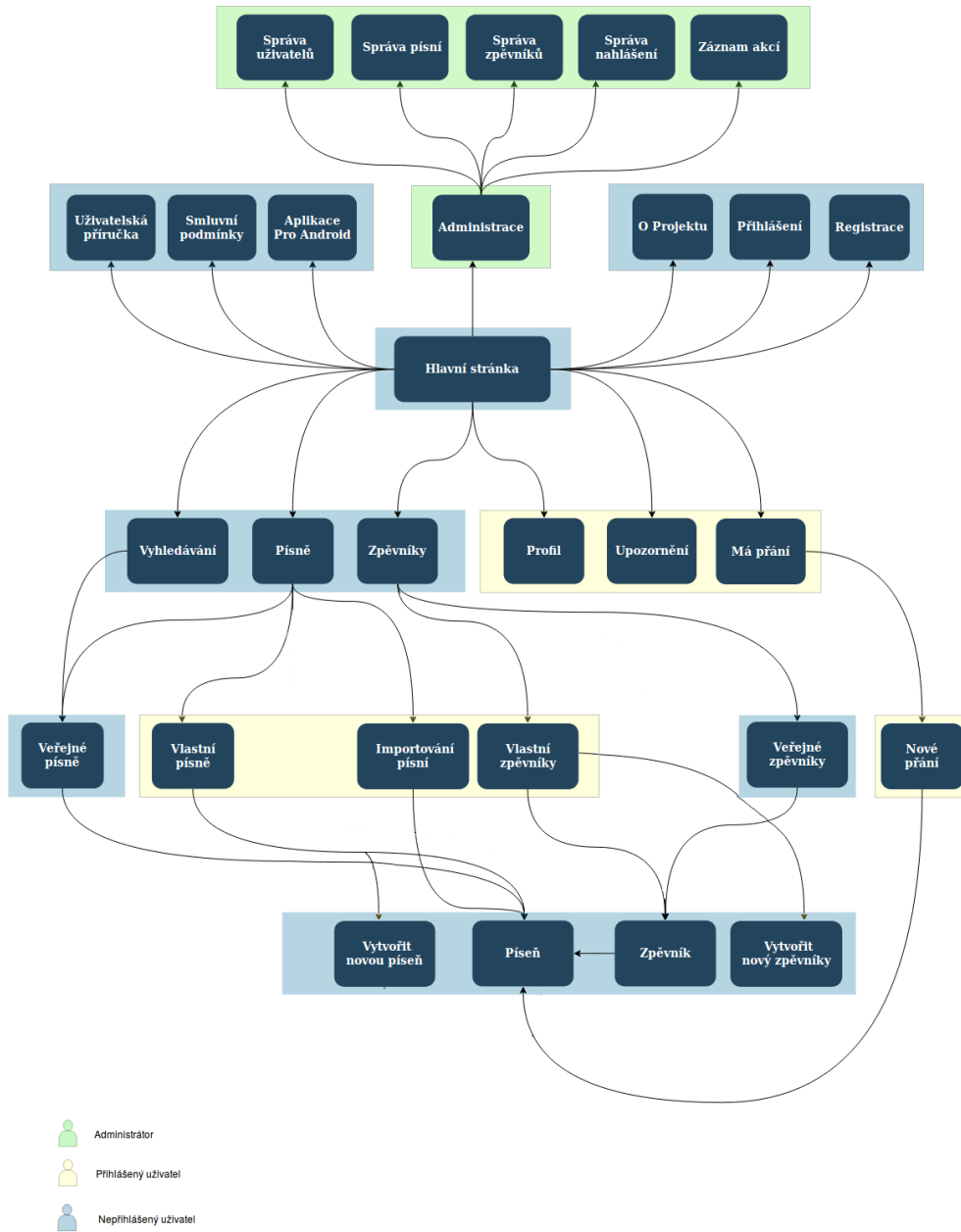
- stránky mají jednoduchou provázanost mezi sebou;
- nejsou zbytečné stránky (např. stránky, jediným účelem kterých – potvrzení nějaké akce);
- odkazy jsou správně pojmenovány (nejsou odkazy s názvem „Klikněte sem“ atd.);

Jediným zaznamenaným nedostatkem je tlačítko pro vytvoření nové písničky nebo nového zpěvníku. Tato akce by měla být dostupná v navigačním menu, protože se jedná o jednu z hlavních funkcí aplikace.

### 1.3 Analýza procesů v aplikaci

Po celkovém auditu analytické dokumentace bylo zaznamenáno více než 20 základních obchodních procesů. Níže v textu jsou uvedeny klíčové a taktéž nově





Obrázek 1.4: Informační architekturu aplikace

aplikované obchodní procesy. Pro jejich názorné zobrazení byly použity diagramy aktivit UML (Unified Modeling Language).

### 1.3.1 Vytvoření písně

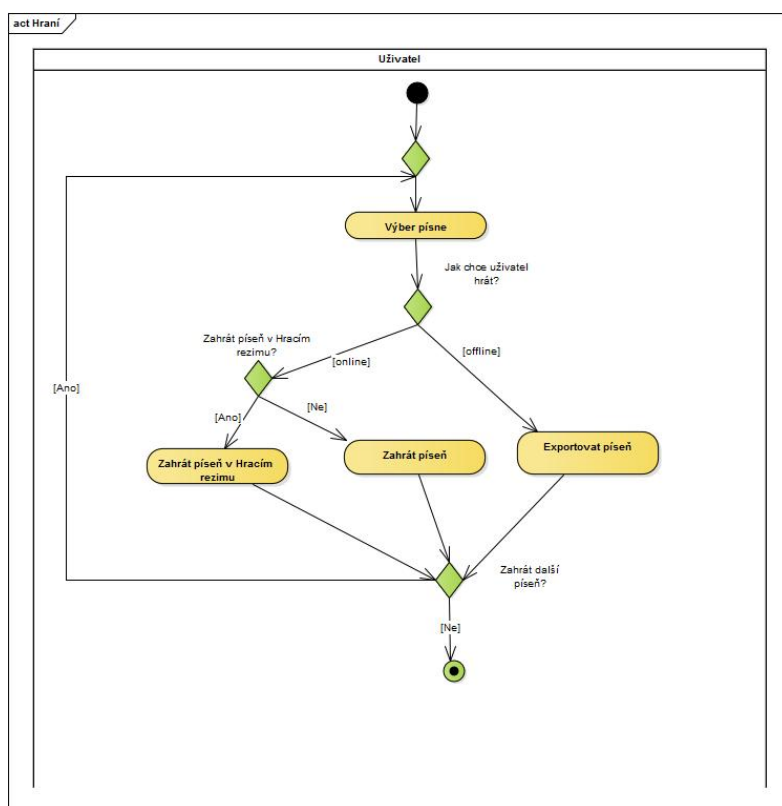
Tento obchodní proces je jedním z nejdůležitějších. Je nezbytné vzít do úvahy fakt, že i prostřednictvím velkého počtu správců obsahů, kteří by vytvářeli platformu slavných písní (dělá to tak část konkurence) by bylo stále nemožné uspokojit potřeby všech uživatelů. Každý uživatel je, co se týká uspokojování jeho potřeb, individuální, a nelze zapomínat na skutečnost, že hraní na kytaru je často doprovázeno projevem tvořivosti. Počínaje příkládáním nových akordů slavných písní a konče vytvořením zcela nových. Hlavně tyto faktory ovlivnily vytvoření tohoto obchodního procesu. Uživatel může narazit na píseň, která se ještě v aplikaci Kytarový zpěvník nenachází, nebo může mít snahu vytvořit vlastní píseň. Vlastní píseň může vytvořit na portálu, v případě potřeby ji může publikovat za účelem získání zpětné vazby od ostatních uživatelů, a co je nejdůležitější, vždycky si ji může zahrát prostřednictvím aplikace.

### 1.3.2 Vytvoření zpěvníku

Jak lze usoudit z názvu aplikace, hlavní myšlenkou je nejen schopnost vytvářet písně, ale taktéž zpěvníky. Uživatel si může písně seskupit podle kritérií, která považuje za důležitá. Může například vytvořit zpěvník, v němž se budou nacházet písně vhodné pro hraní při táboráku nebo si vytvořit zpěvník s písněmi, které by se chtěl naučit hrát. Z dané skutečnosti lze vyvodit závěr, že ta samá píseň se současně může vyskytovat v několika zpěvnících.

### 1.3.3 Přehrát píseň

Bez ohledu na to, že vytváření písní a jejich organizování do zpěvnících jsou velmi důležitými procesy, jedná se o to, že klíčovým procesem při používání aplikace Kytarový zpěvník je proces „Přehrát píseň“. Proto je nezbytné proces přepracovat za účelem vylepšení uživatelského prožitku. Nové řešení bude obsahovat další možnosti, a to: přehrát píseň ve zvláštním režimu. Účelem tohoto procesu je pomoci uživateli soustředit se na hru. V tomto režimu uživatel uvidí pouze text a akordy skladby. Žádné jiné prvky ani obsahová seskupení neodvádějí jeho pozornost. Díky tomuto režimu je píseň zobrazena v aplikaci tak, aby se co nejvíce přibližovala knižní verzi zpěvníku. Aktualizovaný proces je zobrazen na obrázku 1.5.



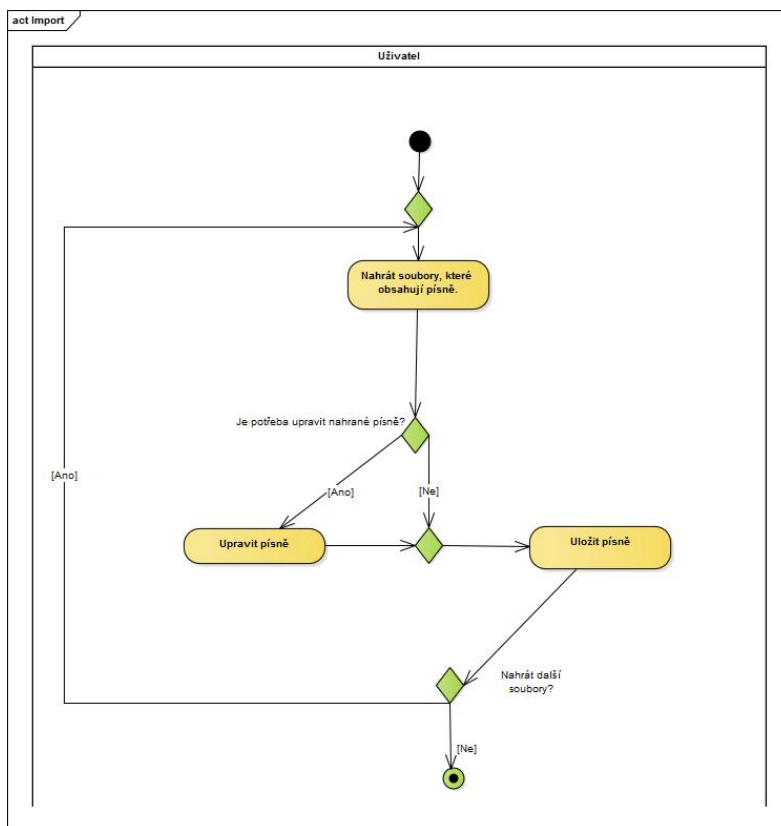
Obrázek 1.5: Obchodní proces hraní písňe

### 1.3.4 Exportovat

V bodě 1.3.2 byl uveden příklad zpěvníku, v němž jsou seskupeny písňe vhodné pro hraní při táboráku. Tento příklad představuje motivaci, proč byl tento obchodní proces vytvořen. Uživatel nevdzdycky má po ruce přístup k internetu nebo zařízení, prostřednictvím něhož by mohl vstoupit do aplikace. Proto je potřeba umožnit mu, aby mohl své písňe kdykoliv převést do papírové podoby.

### 1.3.5 Importovat

Hlavní zdroj, co se týká naplnění portálu obsahem, představují uživatelé. Tento přístup má velkou nevýhodu – osud celého produktu závisí na motivaci uživatelů vytvořit obsah. Proto je nevyhnutelné co nejvíce zjednodušit tvorbu obsahu. Je také třeba přemýšlet nad možností, že někteří uživatelé již podobný produkt používají, ale chtěli by částečně nebo plně přejít do aplikace Kytarový zpěvník. Tento obchodní proces je znázorněn na obrázku 1.6. Byla řešena skutečnost umožnit uživatelům importovat písňe. Uživatel může importovat jednu nebo více písňí najednou. Před uložením má možnost změnit obsah písňe nebo další informace o ní.



Obrázek 1.6: Obchodní proces importovat

## 1.4 Funkční požadavky

### 1.4.1 Odpovídání na komentář

Ve stávajícím řešení má registrovaný uživatel možnost ponechat si komentáře vyplývající ze zpětné vazby. Byl vytvořen nový funkční požadavek, který tuto možnost rozšiřuje. Nyní mohou uživatelé na zprávy ze zpětné vazby reagovat.

### 1.4.2 Vytvoření uživatelské příručky

Uživatelská příručka je manuál, v němž se popisuje správný sled činností potřebných k dovršení jakéhokoli procesu. Vytvoření takového manuálu je časově velmi náročný proces, i z toho důvodu, že je nutné provést úplnou kontrolu všech procesů v aplikaci.

Při vytváření je nezbytné zvážit, s jakými problémy se může uživatel při používání setkat. Například odstranění uživatelského účtu nelze považovat za obchodní proces, ale informace o tom musí uživateli být poskytnuta. Bylo rozhodnuto, že uživatelská příručka bude implementována ve formě často kladených otázek (Frequently Asked Questions, FAQ), to znamená jako seznam

otázek a odpovědí na ně. Vytvoření uživatelské příručky je nepostradatelnou součástí dobrého UX[10].

### 1.4.3 Vytvoření hracího režimu

Účelem zvláštního režimu je oddělit texty a akordy od veškerého jiného obsahu, což má význam z hlediska toho, že při hře na kytaru není rozptylována pozornost uživatele jinými faktory. Funkce automatického posouvání stránek je také přenesena do tohoto režimu. Hlavním důvodem přenosu je skutečnost, že při využívání automatického posouvání vyvstává potřeba neustálého sledování, protože v aktuální verzi se stránka automaticky posouvá na konec stránky, a na konci po hlavním bloku s textem vstupují i komentáře.

### 1.4.4 Import písně

Import písně nebo několika písní musí být proveden automaticky. Uživatel potřebuje pouze vybrat soubory, které jsou uloženy v jeho zařízení. Potom systém zpracuje data těchto souborů a vyzve uživatele, aby je uložil do aplikace. Import musí být implementován ve formátu `.chordpro`[11].

## 1.5 Klíčové aspekty při navrhování UI z hlediska UX

Samozřejmě není možné správným způsobem stanovit kritéria, jejichž dodržování může zajistit úspěch jakéhokoli produktu. Úspěch v první řadě závisí od toho, do jaké míry produkt naplňuje cíle, požadavky na plnění úkolů ze strany uživatele a očekávání uživatelů. Aby bylo možné všechny tyto skutečnosti zaznamenat, je nutné provést hloubkovou analýzu publika, kterému je produkt určen. Na základě získaných dat rozpracovat zobecněné modely uživatelských znaků, identifikovat klíčové skupiny těchto uživatelských znaků a v souladu s předchozím stavem v rámci uživatelského prožitku navrhnout obdobný produkt. Avšak existují základní principy, jejichž dodržování uživatelský prožitek spojený s používáním produktu výrazně vylepší. Pro aplikaci Kytarový zpěvník byly vybrány následující kritéria [12].

### 1.5.1 Požadavky na obsah webu

- text obsahu by měl mít informativní charakter a měl by být vhodně strukturován. Při strukturování textu je vhodné použít dobře zavedený model F-pattern. Podrobněji o této problematice je uvedeno v oddílu 2.2;
- není vhodné vytvářet tzv. „hello“ textové bloky typu „Vítejte“. Tento text neobsahuje žádnou informační hodnotu, pouze marní uživatelův čas;

- v případě používání slideru (několik pravidelně měnících se obrázků v jednom rámečku, které tvoří tzv. slideshow, neboli prezentaci), uživatel by měl dostat následující možnost: zastavit prezentaci, aby si přečetl informace a vrátil se k předchozímu snímku;

### 1.5.2 Typografie

- je nutné dbát na jednotnost textového a barevného řešení;
- při vytváření textu je vhodné použít maximálně dva typy písma;
- není vhodné používat na vytváření textu černou barvu, lze ji nahradit odstíny šedé. Může se zdát, že tento bod není podstatný, ale např. skupina uživatelů s dyslexií určitě ocení šedou barvu[13];
- modrá barva textu by měla být použita pouze v případě zobrazování odkazů;
- pro zobrazení hlavního textu se doporučuje používat velikost písma 16 pixelů [14];

### 1.5.3 Požadavky na strukturu

- navigace by měla uživatelům zpřístupnit důležité informace o webu;
- tlačítko prohlížeče „Zpět“ by mělo vždycky uživatele vrátit na předchozí stránku;
- aplikace by neměla obsahovat blokovací (bezvýchodné) stránky, to jest takové, z nichž se lze, za účelem následného používání aplikace, dostat jen jediným způsobem, a sice kliknout na tlačítko prohlížeče „Zpět“;

### 1.5.4 Požadavky na prvky

- ikony a grafické prvky by měly být uživatelům dobře známé;
- pokud je potřeba vizuálně rozdělit stránku na bloky obsahu, lze použít speciální techniku, která spočívá ve vytvoření kontrastu mezi barvou pozadí a barvou bloku obsahu;

### 1.5.5 Usability

- při používání formulářů, v nichž je zadáno heslo, je třeba dát uživateli možnost skrýt nebo zobrazit obsah těchto formulářů;
- jako dobrá taktika se ukázalo taktéž informování uživatele o zapnutí klávesy „CapsLock“ ve formulářích pro zadávání hesla;

- uživatel by měl být vždycky upozorněn, pokud by se mohl na základě jeho akce obsah smazat;
- zpracování vyplněného formuláře by mělo být spuštěno nejen stiskem výsledného tlačítka, ale také stisknutím klávesy Enter;
- tlačítko tabulátoru musí přesunout kurzor do pole formuláře;
- chybové zprávy by měly být vizuálně rozlišeny (zdůrazněny);
- všechna upozornění by měla být umístěna v horní části stránky;
- při pohybu nad aktivními prvky uživatelského rozhraní by se jejich zobrazení mělo měnit;
- pokud je systémový proces spuštěn více než několik sekund, je nutné o tom informovat uživatele. To se obvykle provádí pomocí animace;

## 1.6 Architektura aplikace a použité technologie

### 1.6.1 Architektura

Jak je znázorněno na obrázku 1.7, aplikace jasně odděluje backend server od klientské části (frontend). Klientská část se spouští na zařízení uživatele, serverová část se nachází na samostatném fyzickém uzlu. Architektonický styl interakce mezi klientem a serverem představuje REST API. Základem komunikace jsou data ve formátu JSON.

Klientská část stránky je vytvořena ve stylu jednostránkové aplikace (Single Page Application, SPA).

K dnešnímu dni existují dva hlavní typy technologií pro vývoj webových aplikací – SPA a MPA (Multi Page Application).

SPA představuje koncepci jednostránkové webové aplikace za předpokladu použití architektonického řešení „Tlustý klient“. Specifičnost „Tlustého klienta“ v tomto případě spočívá v tom, že všechny stránky jsou vygenerovány prostřednictvím jazyka Javascript v klientovi a pouze hlavní údaje o akcích jsou odesílány na server, kde v roli odpovědi přicházejí potřebná data pro dosažení uživatelského cíle. Tato skutečnost má význam v tom, že snižuje závislost na kvalitě komunikace mezi klientem a serverem. Avšak tato metoda má několik podstatných nevýhod:

- část aplikace frontend je velmi komplikovaná. Kromě HTML a logiky UI je nutné vytvořit takzvané směrovače (routery), které napodobují přechod uživatele po stránkách. Navíc obchodní logika aplikace je částečně nebo úplně duplikována;
- vzhledem k tomu, že aplikace má jeden vstupní bod, existuje riziko, že jedna chyba může vést k nefunkčnosti celé aplikace;

- delší doba stahování webu při první návštěvě, protože je potřeba stáhnout celou aplikaci;
- SPA komplikuje optimalizaci webu pro vyhledávače (Search Engine Optimization, SEO). Vzhledem k tomu, že většina webových stránek je vybudována na straně klienta, vyhledávací roboty vyhledávače (boty) vnímají stránku úplně jiným způsobem než uživatel. Tato skutečnost je nejkritičtější pro internetové obchody, blogy a portály, protože při zadání příkazu do vyhledávačů výsledek nebude nalezen.

MPA je tradiční webová aplikace. Funguje na principu, že pokaždé, když aplikace požaduje data nebo je odešle na server, je nucena přijímat novou stránku v plném rozsahu a poté ji zobrazit v prohlížeči. Podrobněji je koncept aplikací MPA popsán v oddílu 2.1.

Kromě výše zmíněných slabin SPA stojí za zmínku fakt, že aplikace Kytařový zpěvník je podstatně zranitelný v oblasti bezpečnosti. Ke kódu, který UI dává k dispozici administrátorům, má přístup kterýkoliv uživatel. Například i neregistrovaný uživatel může zjistit, na kterou adresu administrátor odešle příkaz zablokovat uživatele. Navíc obchodní logika je prakticky v celé míře zkopírována z backend-u, hacking aplikace v tomto případě je jen otázkou času.

### 1.6.2 Použité technologie

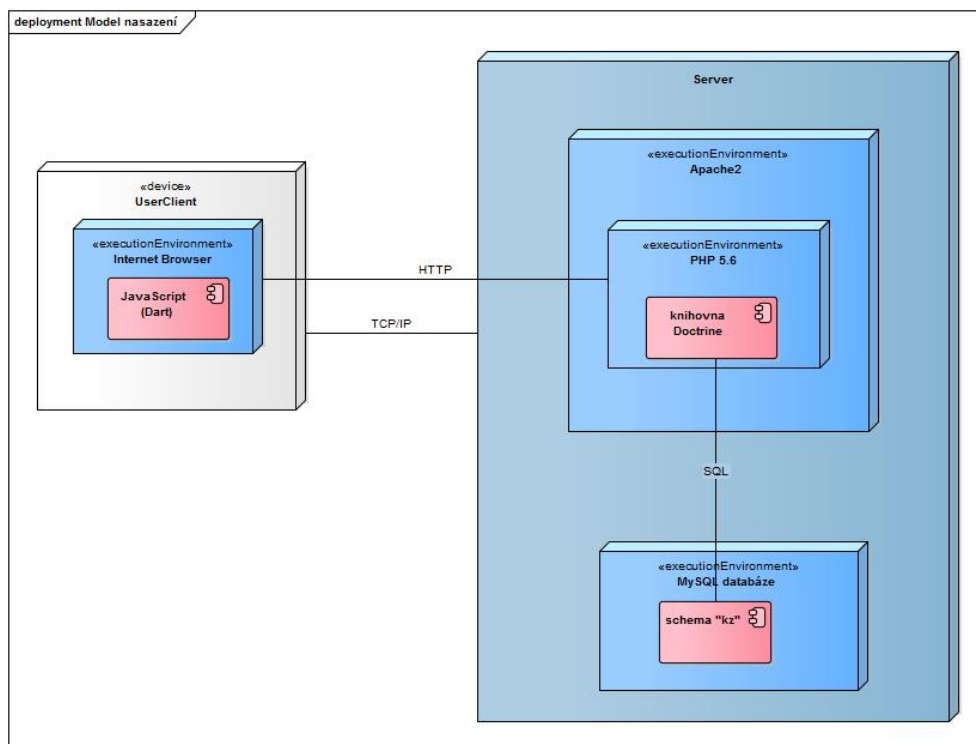
Jako prostředek implementace klientské aplikace byl vybrán programovací jazyk Dart[15]. Dart je relativně nový objektově orientovaný jazyk pro vytváření strukturovaných webových aplikací. Má mnoho výhod, z nichž za hlavní lze považovat syntaxi a strukturu kódu, které jsou velmi podobné již klasickým programovacím jazykům: Java, C++, C#. Grafické uživatelské rozhraní aplikace je implementováno prostřednictvím kaskádních stylů CSS a jazyka JavaScript za účasti knihovny Bootstrap[16]. Další podrobnosti o těchto technologiích jsou popsány v kapitole 3.1.

Za účelem zjednodušení vývojového procesu používá projekt systém správy balíčků, který umožňuje získat mnoho knihoven a softwarových platforem vytvořených různými vývojáři. Avšak, při analýze aplikace bylo zjištěno, že navzdory používání správce balíčků `dart pub` některé knihovny nakonec stále byly nainstalovány manuálně. Například knihovna `bootstrap` je uložena v adresáři projektu. Toto řešení je z hlediska moderního vývoje aplikací nepřijatelné. Kromě manuální aktualizace knihoven musí vývojář identifikovat a manuálně vytvořit všechny potřebné souvislosti.

Sestavení projektu je realizováno na primitivní úrovni. Kromě nevyhnutelnosti kompilace jazyka Dart prostřednictvím `dart-t-js`, je nevyhnutelné manuálně změnit, byť malou, ale přece jen, část kódu. V průvodci pro vývojáře, který je umístěn v hlavní složce projektu, je napsáno:



## 1.6. Architektura aplikace a použité technologie



Obrázek 1.7: Model nasazení

„... Vývoj aplikace je možné realizovat při použití Dartia či s kompilováním po každé změně. Mezi těmito způsoby vývoje se přepíná zakomentováním příslušných řádků v index.php.“

Takové řešení obsahuje řadu problémů, z nichž nejdůležitější jsou:

1. Složitá implementace průběžné integrace (Continuous Integration, CI) projektu. Z hlediska automatizace je potřeba vytvořit další systémové skripty, které budou požadované řádky měnit. Toto je v rozporu se samotnou ideologií CI, jmenovitě: testování i sestavení projektu by mělo být provedeno nad obdrženým zdrojovým kódem (tedy ne nad upraveným kódem).
2. Existuje riziko získání zdrojového kódu v konečné verzi, což může výrazně snížit bezpečnost aplikace.



---

# Návrh

## 2.1 Architektura

Jak již bylo zmíněno v oddílu 1.6.1, existují dva typy webových aplikací – SPA a MPA. Na základě provedené analýzy lze dospět k závěru, že pro aplikaci Kytarový zpěvník použití SPA technologie není optimální.

Proto bylo rozhodnuto aplikaci převést na technologii MPA, která v sobě zahrnuje použití architektonického řešení „Tenký klient“.

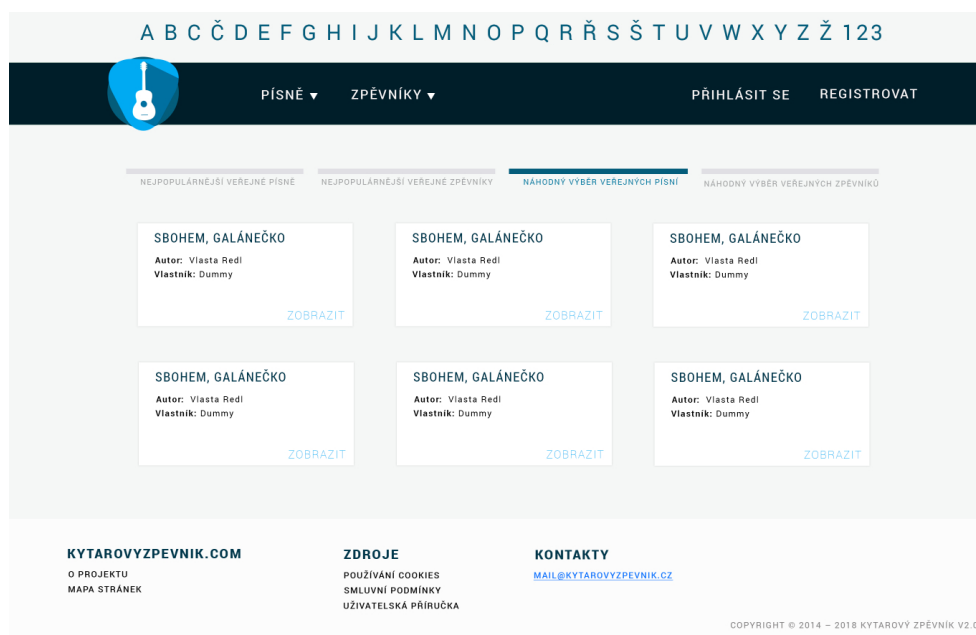
V tomto případě to znamená, že veškerá obchodní logika probíhá na straně serveru. A při každém zadaném požadavku se vygeneruje úplně nová stránka. Avšak generování celé stránky nese s sebou jednu velkou nevýhodu. Každá, byť i nepatrná změna způsobí, že se v prohlížeči zobrazí nová stránka. Například při vytváření komentáře na stránce písně se celá stránka znovu načte. Píseň se při tomto vůbec nemohla změnit, takže restart stránky je nadbytečný. Pro řešení takových případů existuje speciální technologie – Ajax, která umožňuje aktualizovat pouze část stránky. Byla použita tato technologie v součinnosti s využíváním následujících funkcí:

- vytvářet, odstraňovat a upravovat komentáře a hodnocení
- vyhledávat písně a zpěvníky
- přidávat vlastní tagy u písně nebo zpěvníku
- nahlášení veškerého obsahu.

## 2.2 Vytvoření prototypu a návrh designu

Další etapa zahrnuje vytvoření počátečního návrhu nového systému. Jako základ k této činnosti poslouží výsledky provedené analýzy aktuální verze systému a určení kontextu používání a požadavků co se týká nově vzniklého systému.

## 2. NÁVRH



Obrázek 2.1: Prototyp hlavní stránky webu

Z hlediska výběru prototypů byl vyvinut prototyp vysoké přesnosti (tzv. pixel perfect prototype). Vytvořený prototyp je vlastně prototypem rozhraní navrhovaného systému, je vytvořený v grafickém editoru Photoshop[17] a propojený na základě odkazů prostřednictvím speciální webové služby InvisionApp, která umožňuje vytvářet interaktivní prototypy aplikací.

V tomto prototypu však nejsou implementovány všechny funkce, nicméně vyvinuté řešení poskytlo obecnou představu o tom, jak by systém mohl fungovat a vypadat.

Obrázek 2.1 udává prototyp hlavní stránky webu.

Všechny prototypy lze najít na stránce: <https://invis.io/4FFPWGV9EKN>

### 2.3 Návrh řešení požadavků, které vznikly při analýze UI z pohledu UX

Struktura stránky bude implementována pomocí základní šablony F-pattern. F-pattern popisuje nejběžnější stereotyp pohybů očí uživatele při prohlížení stránky. Tato šablona byla popularizována na základě prováděné studie o sledování pohybu očí podle NNGroup[18]. Studie zkoumala chování více než 200 uživatelů, kteří sledovali tisíce webových stránek, na základě čehož se zjistilo, že základní způsob čtení, i když se týkal velkého počtu stránek, byl poměrně konzistentní i z hlediska plnění různých úkolů. Obrázek 2.2 udává, jak uživatelé vnímají stránku webu. Tento vzorek čtení se podobá písmeni F

### 2.3. Návrh řešení požadavků, které vznikly při analýze UI z pohledu UX



Obrázek 2.2: F-pattern šablona[19]



Obrázek 2.3: Nový logotyp aplikace

a je složen z následujících tří součástí:

1. uživatelé nejdříve čtou ve vodorovném směru, obvykle horní část oblasti obsahu. Tento počáteční prvek představuje horní linii písmene F;
2. poté sledují svislou čáru na levé části obrazovky a hledají body zájmu v první větě odstavce. Když objeví něco zajímavého, čtou blíže ve druhém horizontálním směru, který obvykle pokrývá kratší oblast než předchozí směr. Tento přídatný prvek tvoří dolní linii písmene F;
3. nakonec si uživatelé prohlížejí levou stranu obsahu ve svislém směru.

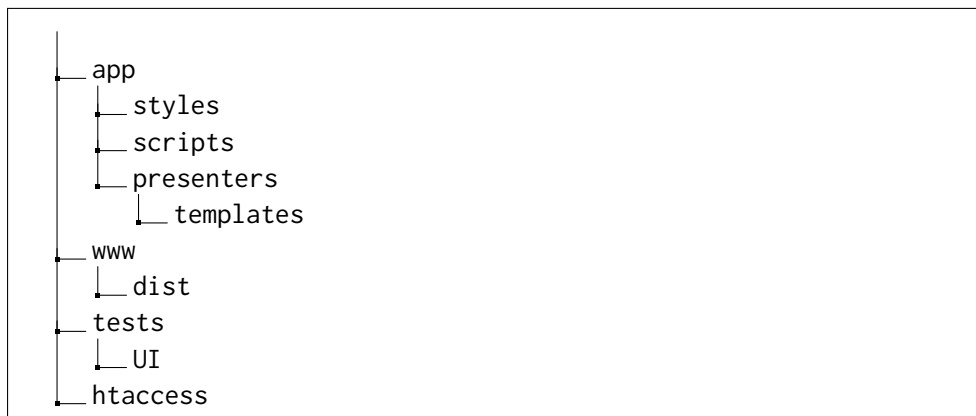
Použití této šablony (z hlediska vývoje aplikace) umožní uživatelům rychle najít užitečné informace.

Po vytvoření prototypů bylo možné konkretizovat následující aspekty návrhu UX:

- v návrhu byl použit Google Material Design[20]. Jeho hlavní předností je rozdělení veškerého textového obsahu do samostatných zón;
- co se týká písma, byl vybrán typ písma Roboto. Tento typ písma je vhodný pro zvolený design a je snadno čitelný;
- hlavní barvy stránky jsou tmavě modré a šedé;
- vytvoření prezentace na domovské stránce představuje dobrou prezentaci produktu pro nové zákazníky. A přidání nových kategorií (nejoblíbenější) může zvýšit motivaci uživatelů k vytváření obsahu.

Při vytváření prototypů bylo navrženo taktéž několik variant logotypů, z nichž jeden byl vybrán. Vybraný logotyp je zobrazen na obrázku 2.3.

Logotyp byl vyhotoven prostřednictvím vektorové grafiky v aplikaci Adobe Illustrator[21]. Při vytváření logotypů nejlepší strategií je držet se minimalistického designu. Tom Wishart, expert na tvorbu logotypů, radí vytvořit logotyp, který vytvoří základní představu o produktu, dále radí co nejvíce se vzdát množství detailů, a přitom zachovat základní myšlenku. Wishart zdůrazňuje, že dobře navržený logotyp má vliv na uživatelský prožitek.[22]



Obrázek 2.4: Struktura projektu

Vytvořený logotyp se vyznačuje proporcionálními tvary a barevnou škálou složenou ze 4 barev. Zobrazuje dva předměty – trsátko a kytaru. Logotyp bez působení kontextu aplikace by měl v uživateli vyvolávat pocit jednoduchosti a pohodlí.

## 2.4 Adresářová struktura projektu

Další krok pozůstával z vytvoření správné struktury projektu. Výsledná struktura klientské části je znázorněna na obrázku 2.4.

- Ve složce `templates` se nacházejí šablony HTML stránek (oddíl 3.1.3);
- ve složce `styles` se nacházejí soubory se styly (oddíl 3.1.4);
- ve složce `tests/UI` se nacházejí testovací skripty (oddíl 3.3);
- ve složce `scripts` se nacházejí soubory JavaScript (oddíl 3.1.5);
- složka `dist` uchovává soubory, které jsou výsledkem programování aplikace, a které budou přímo používány uživateli; (oddíl 3.1.8);
- soubor `.htaccess` představuje soubor zodpovědný za konfiguraci webového serveru a umožňuje tato nastavení používat pro jednotlivé složky na serveru.





---

# Realizace

## 3.1 Použité technologie

### 3.1.1 Konfigurace projektu

Před bezprostřední implementací bylo nutné nakonfigurovat nový projekt. Pro komfortní vývoj byly stanoveny následující požadavky:

- instalace nových balíčků a řešení závislostí musí probíhat automaticky;
- projekt by měl být sestaven automaticky;
- při vývoji je nezbytné použít speciální režim, který bude detekovat změny ve zdrojovém kódu a automaticky sestaví projekt.

### 3.1.2 Správa balíčků

Jak již bylo napsáno v oddílu 1.6.2, pokud neexistuje automatický přístup k instalaci, musí vývojář instalovat balíčky ručně. Pro automatickou instalaci byl vybrán správce balíčků NPM[23].

### 3.1.3 Latte šablony

Zobrazení dat uživateli je realizováno prostřednictvím šablon jazyka Latte[24]. Latte je speciální jazyk vytvořený pro framework Nette[25]. Z hlediska použití těchto šablon byly aplikovány dva přístupy:

1. Kostra stránek byla vytvořena pomocí klasických HTML tagů.
2. Pro naplnění HTML tagů daty byly použity speciální příkazy – Latte makro. Příklad využití je ve výpisu zdrojového kódu 3.1. Z hlediska architektury MPV (Model-View-Presenter), šablony představují zástupce vrstvy View. Ostatní vrstvy jsou realizovány přes framework Nette.

### 3. REALIZACE

---

```
<form class="import-song-form" role="form" n:name="songsForm">
  <div class="float-right">
    <div class="delete-import-form" title="Smazat" >
      <i class="mdi-close mdi icon-black mdi-18px"></i>
    </div>
  </div>
  {label title}
  <input class="form-control" n:name="title">
  {label author}
  <input class="form-control" n:name="author">
  {label publicTags}
  <input class="form-control" n:name="publicTags">
  <div class="text-right">
    {label public}
    <input n:name="public">
  </div>
  {label lyrics}
  <textarea class="form-control" rows="20" n:name="lyrics">
</textarea>
  <div class="text-right">
    <div data-link="{link Song:view id => 0}"
      class="btn btn-primary btn-xs import-song"
      title="Importovat">Importovat</div>
  </div>
</form>
```

Výpis zdrojového kódu 3.1: Realizace formuláře pro import písně

#### 3.1.4 Kaskádové styly

Webové stránky jsou realizovány pomocí značkovacího jazyku HTML a kaskádovými styly CSS. Problém spočívá v tom, že styly nepracují s proměnnými a téměř všechny prvky návrhu musí být vypočítány ručně. Proto při vytváření stylů byl použit jazyk SASS[26] (Syntactically Awesome Style Sheets) – rozšíření kaskádových stylů CSS. Jednou z klíčových funkcí SASS je skutečnost, že disponuje pravidly, jež usnadňují proces vytváření a úpravy vložených selektorů. SASS je nejvíce perspektivní a nejvíc dynamicky se rozvíjející jazyk, který umožňuje rychle a efektivně vytvářet tabulky stylů, které představují základní prvek současného vytváření klientské části webu.

#### 3.1.5 Javascript a jQuery

Z hlediska vytvoření dynamického obsahu a implementace fungování klientské části webu, je vhodné použít Javascript v součinnosti s použitím dodatečných knihoven. Nejdůležitější z nich jsou jQuery a jQuery UI[27], přičemž jQuery umožňuje provádět všechny funkce vývoje klientských částí webů a výrazně

rozšiřuje a zjednodušuje kód díky funkcím, které jsou v něm vytvořeny; a jQuery UI je vybudován na bázi jQuery a zaměřený na návrh základních prvků uživatelského rozhraní webu (menu, textová pole, galerie, atd.). Používání rámců ve vyvíjení představuje charakteristický rys moderního webu a v současné době je nezbytným prvkem.

### 3.1.6 Bootstrap

Bootstrap je nezávislá sada nástrojů pro tvorbu webových stránek a webových aplikací. Zahrnuje HTML a šablony CSS (nová verze poskytuje i SASS šablony) pro návrh typografie, webového formuláře, tlačítek, štítků, navigačních bloků a dalších komponent webového rozhraní včetně rozšíření jazyka Javascript. Při implementaci byla užívána verze Bootstrap 4.

### 3.1.7 Průběžná integrace

Z hlediska systému řízení projektu, v průběhu vývoje byl použit nástroj GIT[28]. Při implementaci bylo nutné vzít do úvahy skutečnost, že vývoj je prováděn nikoliv jedním vývojářem, ale dvěma. Každý z vývojářů by mohl potenciálně změnit kód programu tak, aby změny ovlivňovaly výkon toho druhého. K vyřešení tohoto problému byl použit princip Continuous Integration, v němž vývojáři pravidelně integrují změny kódu do centrálního úložiště, po čemž probíhá automatické sestavování, testování a spuštění v zvláštním prostředí.

V průběhu vývoje byla používána služba GitLab Continuous Integration a integrační proces probíhal na odděleném serveru v několika fázích:

- Získání zdrojového kódu z úložiště GIT
- Sestavení projektu
- Provedení testů
- Nasazení
- Odesílání zpráv

### 3.1.8 Sestavení projektu

Automatické sestavení projektu bylo realizováno prostřednictvím nástroje Webpack[29]. Během automatického sestavování byly kromě kompilace souborů SASS a JS provedeny následující vylepšení:

1. Rozdělení souborů JS do dvou typů: vendor a bundle. V případě vendor – jedná se o kód postranních knihoven, v případě bundle – se jedná o kód napsaný vývojáři. Základní myšlenka spočívá v tom, že soubory typu bundle se mění mnohem častěji než soubory typu vendor. Proto při

implementaci nové verze aplikace budou muset uživatelé stáhnout pouze část aplikace.

2. Část názvu všech souborů, které byly vygenerovány během sestavování, obsahuje unikátní sadu symbolů. Při sestavování bude název změněn pouze u souborů, jejichž kód se změnil. Toto vylepšení má základ v podobné myšlence jako v bodě 1.
3. Implementace tzv. rozdělení kódu (Code Splitting) umožňuje automatické rozdělení javascriptového kódu do samostatných souborů. Tato technika umožňuje snížit velikost souborů typu bundle. To znamená, že každý z těchto souborů bude automaticky načten pouze na ty stránky, které daný kód skutečně používají.
4. Aktivace pluginů, které snižují velikost vygenerovaných souborů bez toho, aby byla ovlivněna jejich funkčnost. Snížení velikosti probíhá především na základě odstranění mezer a komentářů.

## 3.2 Implementace

Implementace se uskutečnila v několika fázích:

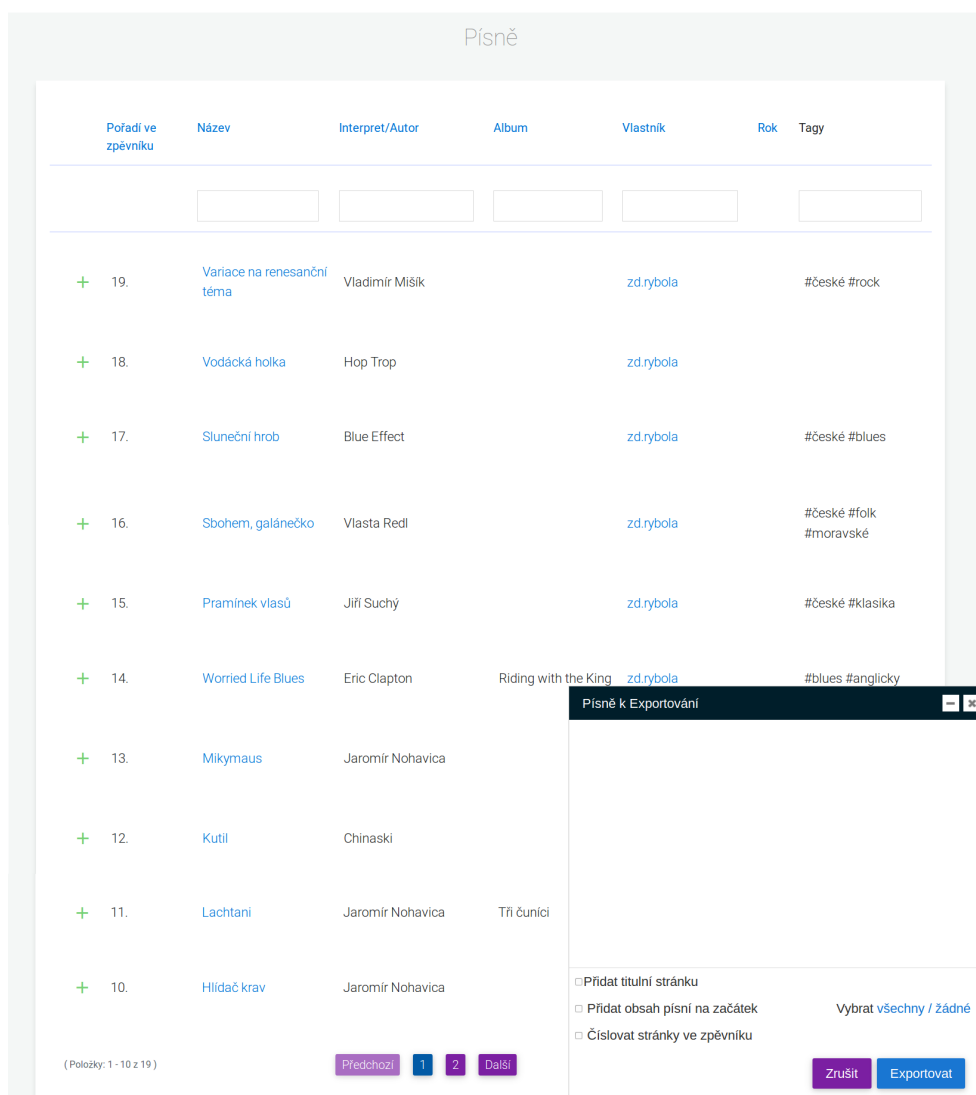
1. vytvoření struktury šablon;
2. vyplnění šablony funkcemi, které implementoval kolega Jan Holan[30];
3. stylizace této šablony s ohledem na koncepci UX, které vyplynuly v rámci prováděné analýzy;
4. aktivace požadovaných prvků prostřednictvím Javascriptu.

Během implementace vzniklo několik nezvyklých situací, které se týkají exportování a importování písně.

### 3.2.1 Exportování písně

Jedno ze zlepšení vyskytující se v nové verzi aplikace Kytarový zpěvník představuje současné zobrazení písní na stránce veřejné/soukromé písně. Dříve byl používán přístup, při němž bylo zobrazeno prvních 20 písní; v případě nutnosti uživatel mohl kliknout na tlačítko „dalších 20 písní“ a systém automaticky načte ještě dalších 20 výsledků. Co se týká exportu, na základě tohoto přístupu výběr písní zvolených uživatelem byl snadno realizovatelný, protože umístění každé zobrazené písně bylo statické. Oproti každé písni byl postaven indikátor, jenž ukazoval, zda píseň při exportu byla brána v úvahu.

V nové verzi se používá jiný přístup, skladby nemají statickou pozici a uživatel nemá vždy možnost vidět, co přesně bude exportováno. Proto bylo



Obrázek 3.1: Dialogové okno pro export zpěvníku

rozhodnuto o implementaci dalšího dialogového okna, které zobrazí aktuální seznam vybraných písní. Bylo také rozhodnuto, že uživatel si může zvolit pozici písně v souboru. Příklad provedení je znázorněn na obrázku 3.1.

### 3.2.2 Importování písně

Hlavní problém s implementací importování písně vznikl při vytváření metody na řešení situace, když je potřeba načíst velké množství souborů. Existují dvě metody:

1. stahování všech souborů na server prostřednictvím jediného požadavku

HTTP;

2. druhá metoda vyžaduje načíst každý soubor samostatně, tj. jeden soubor znamená jeden požadavek HTTP.

Druhá metoda je samozřejmě optimálnější, protože v důsledku asynchronního načítání na server je celková doba provádění snížena. Avšak implementace asynchronního načítání není tak jednoduchá jako vytvoření jediného požadavku, na základě něhož budou všechny soubory přenášeny. Problém s vytvořením asynchronního stahování nastane z toho důvodu, že server po zpracování souboru vrátí informace. Proto je nutné jednak vytvořit několik požadavků, a jednak je také sledovat. Za účelem vyřešení tohoto problému byla znovu použita technologie Ajax. Klíčovou roli při implementaci hraje skrytý HTML tag, zvaný iframe, v němž bude uložen jeden z dočasných souborů, které vybere uživatel pomocí tlačítka „Vyberte soubory“. Javascript detekuje, pokud nějaký z těchto tagů byl změněn (t.j. byl přidán soubor) a text souboru přepošle funkci, která se stará a samotné přeposílání mezi klientem a serverem. Tato funkce je řešena pomocí technologií Ajax. Proto hlavní vlákno prohlížeče nebude čekat, dokud server nevrátí odpověď, ale zkusí najít další iframe tag, ve kterém byla provedena změna.

### 3.3 Testování

Při testování uživatelského rozhraní byla použita speciální testovací metoda nazvaná Regresní testování. Regresní testování je typ testování zaměřeného na kontrolu změn provedených v aplikaci, za účelem získání jistoty, že existující funkce pracují stejně jako předtím.

Provedení testování bylo implementováno prostřednictvím speciálního nástroje – Selenium WebDriver[31].

Selenium WebDriver představuje komfortní API (Application Programming Interface) pro správu prohlížeče, je navržen ve formě knihoven a při testování spouští prohlížeč, který vizuálně zobrazuje všechny odeslané příkazy. WebDriver je schopen pracovat s mnoha prohlížeči a v různých programovacích jazycích. Je snadno integrovatelný s jinými nástroji, protože v podstatě představuje soubor knihoven. Pro tento projekt byl jako prohlížeč vybrán Firefox a jako skriptovací jazyk Javascript. Princip WebDriverů udává diagram znázorněný na obrázku 3.2.

Testy pokrývají zejména hlavní funkčnosti aplikace:

- Autorizace
  - Přihlásit se
  - Odhlásit se
  - Zachovat přihlášení

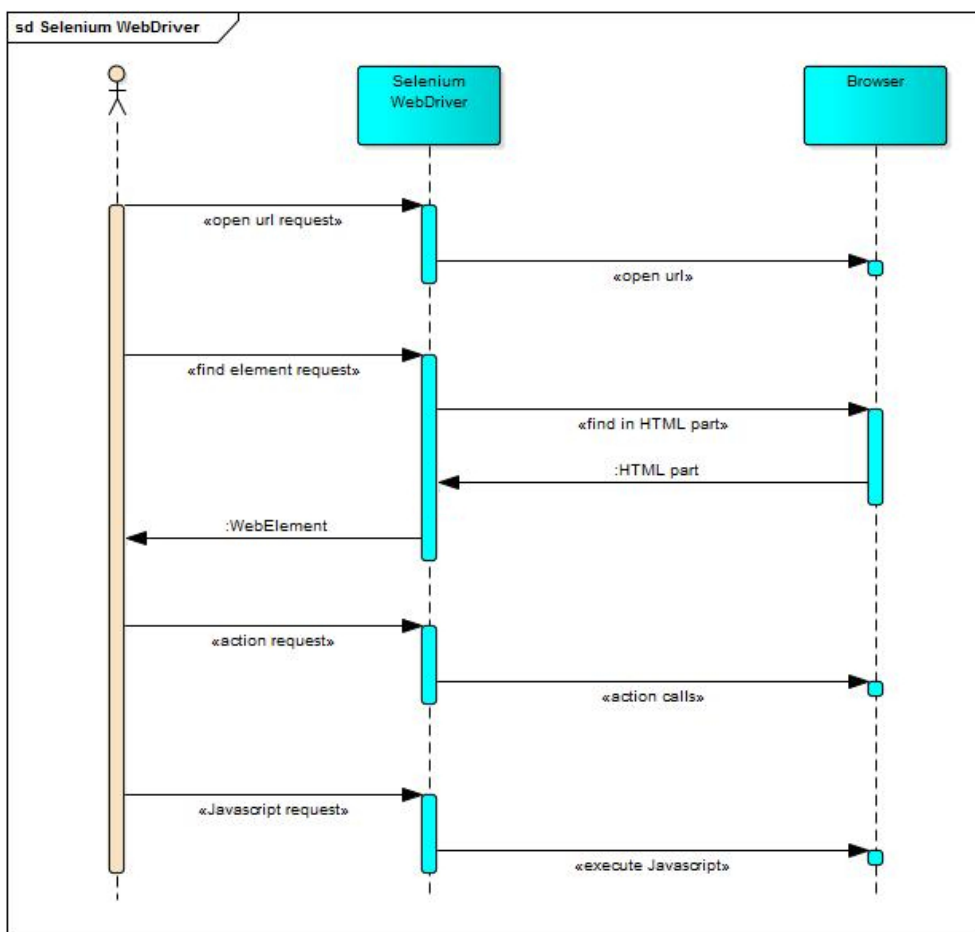
- Píseň
  - Vytvořit
  - Upravit
  - Smazat
- Zpěvník
  - Vytvořit
  - Upravit
  - Přidat píseň
  - Smazat
- Přáním
  - Vytvořit
  - Upravit
  - Smazat
- Komentář
  - Vytvořit
  - Upravit
  - Smazat
- Vyhledávání
  - písně podle názvu
  - zpěvníku podle názvu

### 3.4 Nasazení

Jak již bylo uvedeno výše, při vývoji aplikace Kytarový zpěvník byla použita metodika Continuous Integration. Proto, po provedení implementační práce byla konečná verze téměř připravena, bylo nutné pouze odstranit soubory, které byly použity pro vývoj. V této fázi byla taktéž aktualizována příručka pro vývojáře a průvodce nasazením aplikace.

### 3. REALIZACE

---



Obrázek 3.2: Princip činnosti Selenium WebDriver



---

## Závěr

Práce na sepsání tohoto příspěvku vyžadovala širokou škálu znalostí. Kromě využití znalostí z oblasti vývoje aplikací bylo nutné nejen studovat technologie spojené s User Experience, ale taktéž je použít v praxi. Ještě před začátkem práce bylo jasné, že celá klientská část aplikace bude zcela přepracována, toto rozhodnutí ovlivnilo mnoho faktorů, včetně změny architektury, přesunu obchodní logiky na stranu serveru a změnu jazyka, ve kterém byla klientská část aplikace napsána.

V rámci práce na tomto příspěvku byl zaznamenán standardní životní cyklus aplikace z hlediska vývoje moderního softwaru: analýza, návrh a implementace.

V rámci analýzy bylo nejdříve studováno a popsáno uživatelské rozhraní aplikace, a to jak z hlediska jeho implementace, tak i co se týká hlediska uživatelské prožitku podle User Experience. Následovalo vytvoření designu a jeho prototypu. Taktéž byly identifikovány a postaveny nové funkční požadavky, které byly následně zrealizovány; zvláštní pozornost byla věnována vypracování podrobného manuálu pro uživatele. V poslední fázi vývoje byly provedeny automatické testy a tvorba finální verze produktu.

Výsledek práce na tomto příspěvku představuje aplikace Kytarový zpěvník 2.0. Aplikace se díky implementovaným změnám stala rychlejší, stabilnější a bezpečnější. Momentálně se nachází ve stavu, že skutečně může být používána ve veřejném vlastnictví.



---

## Literatura

- [1] Brooks, F. P.: *The mythical man-month: essays on software engineering*. Estados Unidos: Addison-Wesley, 1995, ISBN 0201835959, 122 s.
- [2] Merholz, P.: Peter in Conversation with Don Norman About UX Innovation. December 2007, [Online; accessed 20 . June 2018]. Dostupné z: <http://adaptivepath.org/ideas/e000862/>
- [3] International Organization for Standardization: *Ergonomics of human-system interaction*. Geneva: BSi, 2010, ISBN 9780580640094, 3 s.
- [4] Morville, P.: User Experience Design. June 2004, [Online; accessed 20 . June 2018]. Dostupné z: [http://semanticstudios.com/user\\_experience\\_design/](http://semanticstudios.com/user_experience_design/)
- [5] Reichenstein, O.: The Spectrum of User Experience. June 2009, [Online; accessed 20 . June 2018]. Dostupné z: <https://ia.net/topics/the-spectrum-of-user-experience-1>
- [6] Garrett, J. J.: *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Berkeley, California: New Riders, druhé vydání, 2010, ISBN 0321683684, 18 s.
- [7] Bhargava, T.: How To Create A Memorable Experience For Your App Users. June 2017, [Online; accessed 20 . June 2018]. Dostupné z: <http://blog.linktexting.com/how-to-create-a-memorable-experience-for-your-app-users/>
- [8] Obias-Tuban, A.: Starting out in UX: What it actually means, when you say that you'll do UX for a living. May 2015, [Online; accessed 20 . June 2018]. Dostupné z: <https://medium.com/the-redesign/starting-out-in-ux-what-it-actually-means-when-you-say-that-you-ll-do-ux-7ed0e45770ef>

- [9] Wong, A.: 30 Days of Product Management Genius with Oz Chen. Introduction to UX for Product Managers. December 2017, [Online; accessed 20 . June 2018]. Dostupné z: <https://medium.com/productquest/30-days-of-product-management-genius-with-oz-chen-introduction-to-ux-for-product-managers-889f9bd5dfa1>
- [10] Farrell, S.: FAQs Still Deliver Great Value. December 2014, [Online; accessed 20 . June 2018]. Dostupné z: <https://www.nngroup.com/articles/faqs-deliver-value/>
- [11] Martin Leclerc, M. D.: ChordPro. [Online; accessed 20 . June 2018]. Dostupné z: <https://www.chordpro.org/>
- [12] Kucheriavy, A.: Top 100 UX Design Tips from a User Experience Master. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://www.intechinc.com/blog/100-ux-design-pro-tips-from-user-experience-master/>
- [13] Gabriel, R. J.: Research into Web Accessibility for Dyslexics and Dyslexia-focused fonts such as OpenDyslexia. February 2017, [Online; accessed 20 . June 2018]. Dostupné z: <https://medium.com/@RobertJGabriel/research-into-web-accessibility-for-dyslexics-and-dyslexia-focused-fonts-such-as-opens-dyslexia-6ff27a10773b>
- [14] Reichenstein, O.: The 100% Easy 2 Read Standard. November 2006, [Online; accessed 20 . June 2018]. Dostupné z: <https://ia.net/topics/100e2r>
- [15] Google LLC: Dart programming language. 2018, [Online; accessed 20 . June 2018]. Dostupné z: [dartlang.org](http://dartlang.org)
- [16] Otto, M.: Bootstrap. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <http://getbootstrap.com/>
- [17] Adobe Systems Incorporated: Adobe Photoshop Learn & Support. June 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://helpx.adobe.com/support/photoshop.html>
- [18] Nielsen Norman Group: F-Shaped Pattern For Reading Web Content (original eyetracking research). April 2006, [Online; accessed 20 . June 2018]. Dostupné z: <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/>
- [19] Mico Yuk, S. D.: Data Visualization: Common Visual Patterns. [Online; accessed 20 . June 2018]. Dostupné z: <http://www.dummies.com/programming/big-data/big-data-visualization/data-visualization-common-visual-patterns/>

- 
- [20] Google LLC: Google's new 'Material Design' UI coming to Android, Chrome OS and the web. June 2014, [Online; accessed 20 . June 2018]. Dostupné z: <https://www.engadget.com/2014/06/25/googles-new-design-language-is-called-material-design/>
- [21] Adobe Systems Incorporated: Adobe Illustrator Learn & Support. June 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://helpx.adobe.com/support/illustrator.html>
- [22] Wilshart, T.: How to use Logo Design to Enhance User Experience. October 2016, [Online; accessed 20 . June 2018]. Dostupné z: <http://digitalmarketingmagazine.co.uk/customer-experience/how-to-use-logo-design-to-enhance-user-experience/3705>
- [23] Reed, N.: What is npm? August 2011, [Online; accessed 20 . June 2018]. Dostupné z: <https://docs.nodejitsu.com/articles/getting-started/npm/what-is-npm/>
- [24] Nette Foundation: Latte - amazing template engine for PHP. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://latte.nette.org/en/>
- [25] Nette Foundation: Rychlý a pohodlný vývoj webových aplikací v PHP — Nette Framework. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://nette.org>
- [26] Giraudel, H.: Syntactically Awesome Style Sheets. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://sass-lang.com/>
- [27] jQuery Foundation: jQuery. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <http://jquery.com/>
- [28] Chacon, S.: Git - A Short History of Git. 2018, [Online; accessed 20 . June 2018]. Dostupné z: <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>
- [29] Larkin, S. T.: webpack 2.2: The Final Release. January 2017, [Online; accessed 20 . June 2018]. Dostupné z: <https://medium.com/webpack/webpack-2-2-the-final-release-76c3d43bf144>
- [30] Jan, H.: *Přepřacování aplikace Kytarový zpěvník do MVP architektury a návrh gamifikace aplikace*. 2018, Bakalářská práce. Fakulta informačních technologií České vysoké učení technické v Praze, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Rybola.
- [31] Stewart, S. M.: Selenium 3 is Coming. October 2016, [Online; accessed 20 . June 2018]. Dostupné z: <https://seleniumhq.wordpress.com/2016/10/04/selenium-3-is-coming/>



## Seznam použitých zkratk

- API** Application Programming Interface
- CI** Continuous Integration
- CSS** Cascading Style Sheets
- FAQ** Frequently Asked Questions
- HCI** Human Computer Interaction
- HTML** Hypertext Markup Language
- HTTP** Hypertext Transfer Protocol
- ISO** International Organization for Standardization
- IA** Information Architecture
- JS** Javascript
- JSON** Javascript Object Notation
- MPA** multiple-page application
- MVP** Model View Presenter
- REST** Representational State Transfer
- SASS** Syntactically Awesome Style Sheets
- SPA** Single Page Application
- SEO** Search Engine Optimization
- UxD** User Experience Design
- UI** User Interface

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**UX** User Experience

**UML** Unified Modeling Language

**WWW** World Wide Web



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
release.....	aplikace určená k nasazení na server
├─ releaseGuide_2.0.pdf.....	instalační příručka
├─ sql.....	počáteční data pro databázi
└─ src.....	zdrojové kódy webových stránek
src	
├─ implementation	
│ └─ devGuide_2.0.pdf.....	programátorská příručka
│ └─ src.....	zdrojové kódy implementace
└─ prototypes.....	prototypy aplikace ve formátu PSD a AI
thesis.....	písemná práce
├─ src.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
└─ thesis.pdf.....	text práce ve formátu PDF